

Raport – analiza oprogramowania Cry

Próbka

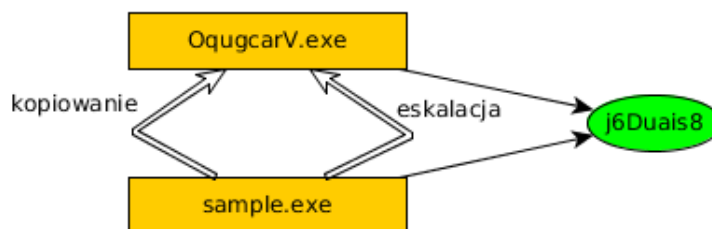
Analizowana próbka to moduł PE32 o następujących skrótach kryptograficznych.

Md5	41ec3fef7b25b3cde15737a27d16735
Sha256	a6a8b82acdf7d2c873c453f740527d33c758a97c29d4f76cad6843118b7cfaaf
Sha512	138dec7a43d46fc2e4cf279feceb20065bf08a83271e6273eefcd909df495cce1f1e d23a8c6ae123c22dacd8ed999c58621fc9c935d6abb2f9b6defdb425dfba

Próbka została uzyskana w trakcie analizy powłamiowej w systemie jednego z klientów Orange Polska.

Architektura

Cry to aplikacja typu ransomware. Jej celem jest zaszyfrowanie potencjalnie ważnych danych na komputerze ofiary i żądanie okupu za klucz deszyfrujący. Ofiara otrzymuje komunikat o żądaniu okupu i w efekcie realizuje transakcję na rzecz cyberprzestępców. Na koniec cyberprzestępca przekazuje instrukcje, co zrobić, by odszyfrować dane.



Cry posiada prostą architekturę – jeden moduł wykonywalny, który sam kopiuje siebie w głąb systemu plików, a następnie usiłuje się uruchomić z podwyższonymi uprawnieniami. Podczas działania korzysta on z mutexa synchronizującego działanie oraz tworzy różne klucze rejestru oraz skróty, z których korzysta w trakcie prowadzenia swoich operacji.

Działanie

Cry najpierw sprawdza, czy infekcja została już przeprowadzona, poszukując mutexu z wygenerowaną pseudolosową nazwą. Jeśli mutex nie istnieje, malware przystępuje do infekowania komputera.

Infekcja ogranicza się do skopiowania podstawowego modułu do wyznaczonej lokacji w systemie plików oraz utworzenia wskazującego na siebie skrótu w folderze Startup. Nowa nazwa modułu jest generowana przez generator pseudolosowy. Po przeprowadzeniu tych operacji podejmowana jest próba uruchomienia skopiowanego w głąb systemu plików modułu z podwyższonymi uprawnieniami.

```

.text:00402880 ;
.text:00402880
.text:00402880 loc_402880:
.text:00402880 lea    eax, [esp+0Ch]
.text:00402884 push  eax
EIP .text:00402885 call  ebx                ; shell132_ShellExecuteExW
.text:00402887 test  eax, eax
.text:00402889 jnz   short loc_4028AF
.text:0040288B mov   ecx, esi
0000287F 0040287F: .text:0040287F

Hex View-1
00298A20 2F 00 43 00 20 00 22 00 43 00 3A 00 5C 00 50 00  /..C..".C.:.\.P.
00298A30 72 00 6F 00 67 00 72 00 61 00 6D 00 44 00 61 00  r.o.g.r.a.m.D.a.
00298A40 74 00 61 00 5C 00 4F 00 71 00 75 00 67 00 63 00  t.a.\.0.q.u.g.c.
00298A50 61 00 72 00 56 00 2E 00 65 00 78 00 65 00 22 00  a.r.U..e.x.e.".
00298A60 20 00 31 00 00 00 AB AB AB AB AB AB AB EE FE  -.1...zzzzzzzz!
00298A70 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  20 00
    
```

Po udanym podwyższeniu uprawnień Cry wysłała raport poprzez kanał C&C, którego konstrukcja została omówiona w dalszej części raportu.

Następnym krokiem jest usunięcie punktów przywracania systemu (shadow volumes) i przystąpienie do szyfrowania potencjalnie cennych plików w systemie ofiary. Klucz wykorzystywany do szyfrowania jest generowany za pomocą systemowych usług kryptograficznych.

```

00401E04
00401E04 loc_401E04:
00401E04 mov   edx, [esp+0C8h+arg_10]
00401E08 lea  ecx, [esp+0C8h+var_B8]
00401E0F push ecx
00401E10 mov  ecx, [esp+0CCh+arg_C]
00401E17 push ebx
00401E18 push ebx
00401E19 push edx
00401E1A push ecx
00401E1B push eax
00401E1C call ds:advapi32_CryptImportKey
00401E22 test  eax, eax
00401E24 jnz  short loc_401E33

00401EAC
00401EAC loc_401EAC:
00401EAC push  edi
00401EAD lea  edx, [esp+0CCh+var_A4]
00401EB1 xor  eax, eax
00401EB3 push edx
00401EB4 push  20h
00401EB6 mov  byte ptr [esp+0D4h+var_AC], bl
00401EBA mov  [esp+0D4h+var_AC+1], eax
00401EBE mov  [esp+0D4h+var_A7], ax
00401EC3 mov  [esp+0D4h+var_A5], al
00401EC7 call sub_4024F0
00401ECC mov  esi, [esp+0D4h+arg_0]
00401ED3 push 20h
00401ED5 lea  eax, [esp+0D8h+var_A4]
00401ED9 push eax
00401EDA push esi
00401EDB call loc_406840

00401EE0
00401EE0 ; LPDWORD loc_401EE0
00401EE0 loc_401EE0:
00401EE0 mov  edi, [esp+0C8h+arg_1C]
00401EE7 push 005h
00401EEC lea  ecx, [esi+20h]
00401EEF push edi
00401EF0 push ecx
00401EF1 call loc_406840
00401EF6 mov  eax, [esp+0D4h+var_A0]
00401EFA add  esp, 20h
00401EFB push 100h
00401F02 lea  edx, [esp+0D8h+var_98]
00401F06 push edx
00401F07 push esi
00401F08 push ebx
00401F09 push ebx
00401F0A push ebx
00401F0B push eax
00401F0C mov  [esp+0D0h+var_98], 0F5h
00401F14 call ds:advapi32_CryptEncrypt
00401F1A test  eax, eax
    
```

Cry posługuje się dwoma parami kluczy RSA. Pierwsza z nich, nazywana dalej kluczami operacyjnymi, jest generowana w procesie infekcji na komputerze ofiary. Jej publiczna część jest wykorzystywana do szyfrowania danych, natomiast część prywatna jest szyfrowana za pomocą publicznej części drugiej pary

kluczy (zwanej dalej kluczami głównymi). Za pomocą szeregi operacji arytmetycznych prywatny klucz operacyjny przekształcany jest do indywidualnego kodu identyfikującego ofiarę.

Tworzona jest również notatka z żądaniem okupu, która jest ustawiana jako tło pulpitu. Aby upenić się, że ofiara zauważy notatkę, wszystkie skróty z pulpitu są przenoszone do katalogu „old_shortcuts”, tak, by jej nie zasłaniały.

Identyfikator ofiary jest przechowywany w rejestrze systemowym i można do niego uzyskać dostęp w każdym momencie. W trakcie wpłacania okupu ofiara za pośrednictwem dedykowanej strony w sieci TOR ma go przekazać botmasterowi, który po potwierdzeniu wpłaty okupu rozszyfruje go prywatnym kluczem głównym, co pozwala odzyskać prywatną część klucza operacyjnego w systemie ofiary i odszyfrować dane.

Szyfrowanie

Przed przystąpieniem do szyfrowania i usuwania niezaszyfrowanych plików aplikacja usuwa punkty przywracania systemu Volume Shadow Copies.

Samo szyfrowanie jest przeprowadzane w dwóch etapach. W pierwszym etapie przeprowadzana jest analiza systemu plików ofiary. Ransomware w rekurencyjnie wykonywanej procedurze przeszukuje strukturę katalogów w poszukiwaniu kandydatów do szyfrowania. Warunkami, które musi spełnić plik jest posiadanie jednego z kilkudziesięciu rozszerzeń, przy czym istnieje również lista nazw plików wykluczonych.

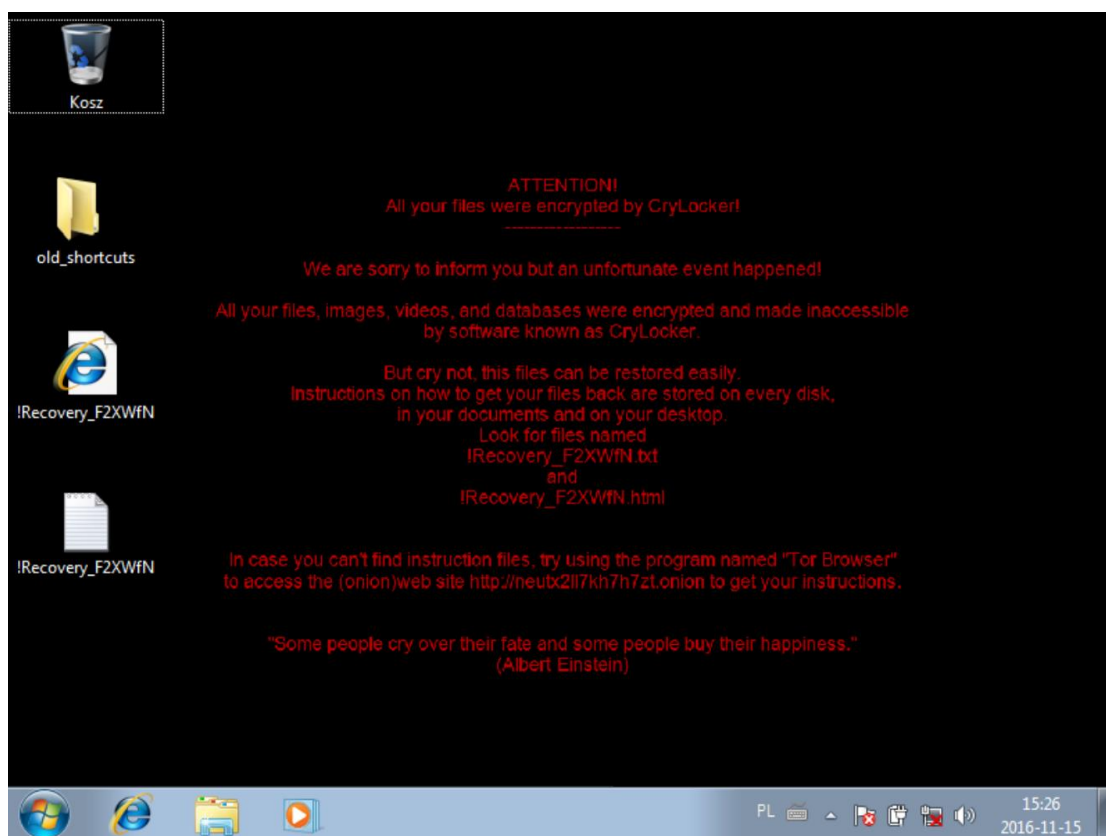
01AD91AC	5C 00 5C 00 3F 00 5C 00	43 00 3A 00 5C 00 50 00	\\.\?.\C.: \.P.
01AD91BC	79 00 74 00 68 00 6F 00	6E 00 32 00 37 00 5C 00	y.t.h.o.n.2.7.\.
01AD91CC	4C 00 69 00 62 00 5C 00	74 00 65 00 73 00 74 00	L.i.b.\.t.e.s.t.
01AD91DC	5C 00 6F 00 75 00 74 00	73 00 74 00 61 00 6E 00	\.o.u.t.s.t.a.n.
01AD91EC	64 00 69 00 6E 00 67 00	5F 00 62 00 75 00 67 00	d.i.n.g._.b.u.g.
01AD91FC	73 00 2E 00 70 00 79 00	00 00 AB AB AB AB AB AB	s...p.y...zzzzzz
01AD920C	AB AB EE FE 00 00 00 00	00 00 00 00 ED A4 12 47	zzt!.....ŸA.G
01AD921C	80 00 0D 1A 00 00 00 00	BD A7 00 00 DD 00 00 00	ç.....žž..J...
01AD922C	5C 00 5C 00 3F 00 5C 00	43 00 3A 00 5C 00 50 00	\\.\?.\C.: \.P.
01AD923C	79 00 74 00 68 00 6F 00	6E 00 32 00 37 00 5C 00	y.t.h.o.n.2.7.\.
01AD924C	4C 00 69 00 62 00 5C 00	74 00 65 00 73 00 74 00	L.i.b.\.t.e.s.t.
01AD925C	5C 00 70 00 69 00 63 00	68 00 6C 00 65 00 74 00	\.p.i.c.k.l.e.t.
01AD926C	65 00 73 00 74 00 65 00	72 00 2E 00 70 00 79 00	e.s.t.e.r...p.y.
01AD927C	00 00 AB AB AB AB AB AB	AB AB EE FE 00 00 00 00	..zzzzzzzzt!....
01AD928C	00 00 00 00 EC A4 12 46	9F 00 0D 1A 00 00 00 00	...ŸA.Fç.....
01AD929C	54 0C 00 00 DD 00 00 00	5C 00 5C 00 3F 00 5C 00	T...J...\\.\?.\.
01AD92AC	43 00 3A 00 5C 00 50 00	79 00 74 00 68 00 6F 00	C.: \.P.y.t.h.o.
01AD92BC	6E 00 32 00 37 00 5C 00	4C 00 69 00 62 00 5C 00	n.2.7.\.L.i.b.\.
01AD92CC	74 00 65 00 73 00 74 00	5C 00 70 00 72 00 6F 00	t.e.s.t.\.p.r.o.
01AD92DC	66 00 69 00 6C 00 65 00	65 00 2E 00 70 00 79 00	f.i.l.e.e...p.y.
01AD92EC	00 00 AB AB AB AB AB AB	AB AB EE FE 00 00 00 00	..zzzzzzzzt!....
01AD92FC	00 00 00 00 ED A4 12 47	9E 00 0D 1A 00 00 00 00	...ŸA.Gx.....
01AD930C	A9 02 00 00 DD 00 00 00	5C 00 5C 00 3F 00 5C 00	e...J...\\.\?.\.
01AD931C	43 00 3A 00 5C 00 50 00	79 00 74 00 68 00 6F 00	C.: \.P.y.t.h.o.
01AD932C	6E 00 32 00 37 00 5C 00	4C 00 69 00 62 00 5C 00	n.2.7.\.L.i.b.\.
01AD933C	74 00 65 00 73 00 74 00	5C 00 70 00 79 00 63 00	t.e.s.t.\.p.y.c.
01AD934C	6C 00 62 00 72 00 5F 00	69 00 6E 00 70 00 75 00	l.b.r._.i.n.p.u.
01AD935C	74 00 2E 00 70 00 79 00	00 00 AB AB AB AB AB AB	t...p.y...zzzzzz
01AD936C	AB AB EE FE 00 00 00 00	00 00 00 00 ED A4 12 47	zzt!.....ŸA.G

Po utworzeniu listy uruchamiane są wątki, które przeprowadzają szyfrowanie wszystkich plików z listy.

Zawartość pliku najpierw kopiowana jest do katalogu tymczasowego i szyfrowana, a następnie plik docelowy jest zastępowany nową, zaszyfrowaną wersją, podczas, gdy plik w katalogu tymczasowym jest usuwany.

Notka

Po przeprowadzeniu tych operacji, malware informuje ofiarę o przeprowadzonych operacjach i możliwości zapłaty okupu, umieszczając na pulpicie pliki zawierające instrukcje kontaktowania się z botmasterem za pośrednictwem portalu w sieci TOR. Dodatkowo przenosi wszystkie pliki z pulpitu do katalogu old_shortcuts, a tło pulpitu zmienia na notatkę z odniesieniem do tych instrukcji.



Dystrybucja kluczy

Aplikacje typu ransomware posiadają nieco inne kanały C&C w stosunku do złośliwego oprogramowania innych typów (np. RAT, bankers, etc.). Wynika to z faktu, że prowadzenie kampanii ransomware nie wymaga wymiany dużej ilości danych pomiędzy botem a botmasterem, zaś większość złośliwych operacji może zostać przeprowadzona bez interakcji pomiędzy nimi.

W przypadku aplikacji ransomware botmaster ma do czynienia tylko z jednym problemem związanym z komunikacją: z generowaniem i dystrybucją kluczy.

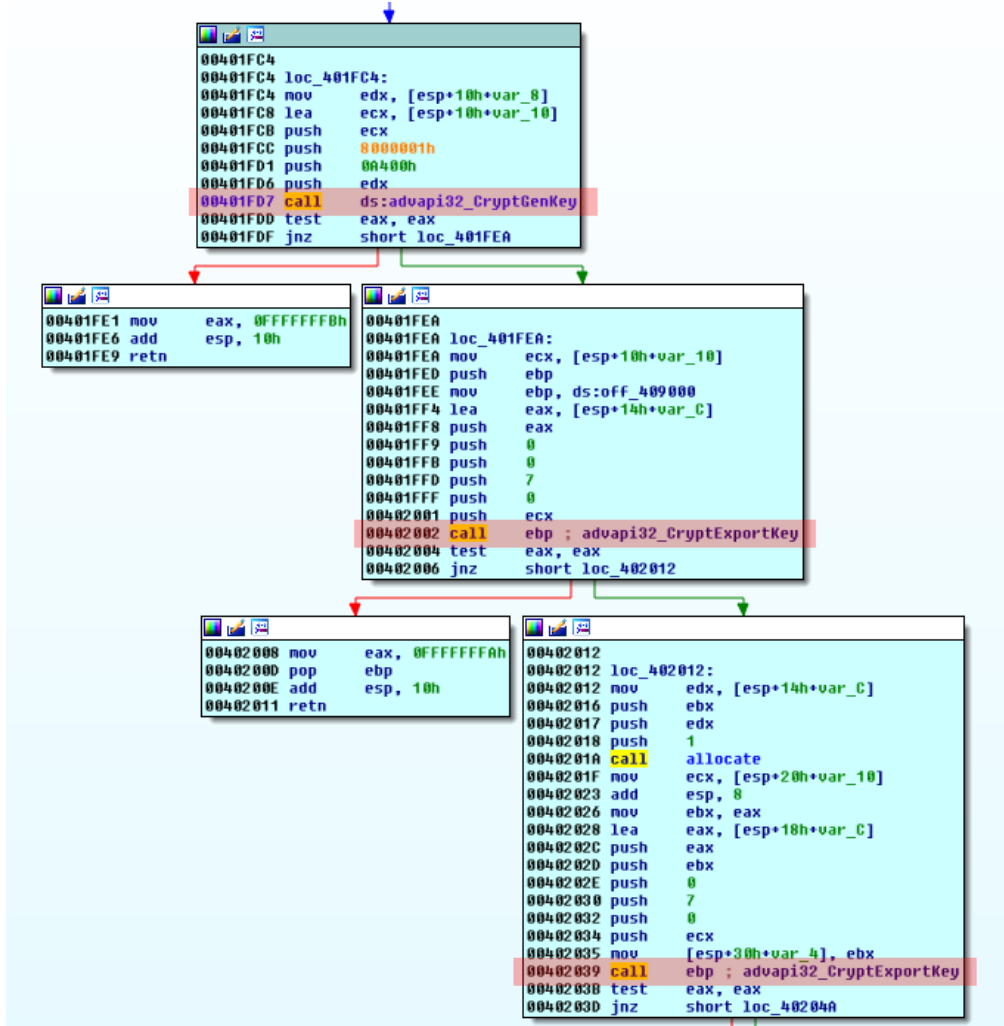
Po pierwsze, w przypadku ransomware kryptografia symetryczna nie nadaje się do zastosowania. Wynika to z tego, że po pozyskaniu próbki w laboratorium podczas analizy procesu szyfrowania można uzyskać klucz i algorytm potrzebny do odszyfrowania danych, a co za tym idzie – do porażki kampanii. Z tego powodu twórcy oprogramowania ransomware opierają się na kryptografii asymetrycznej.

Drugą kwestią, którą należy wziąć pod uwagę w związku z kryptografią jest fakt, że każda para kluczy powinna być wygenerowana indywidualnie dla konkretnej ofiary. Jeśli wszystkie ofiary zostałyby zaatakowane z wykorzystaniem tej samej pary kluczy, mogłoby dojść do sytuacji, gdy jedna z ofiar po wpłacie okupu udostępniła klucz deszyfrujący pozostałym ofiarom. A brak konieczności zapłaty okupu oznacza porażkę cyberprzestępcy.

Z tych powodów aplikacje ransomware generują indywidualne pary kluczy kryptografii asymetrycznej. W części próbek proces ten był wykonywany za pośrednictwem bezpośredniej komunikacji bota z botmasterem.

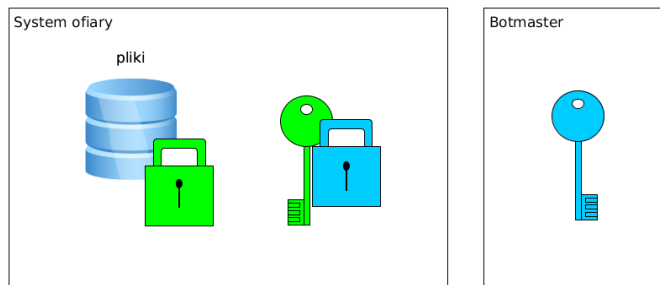
Przykład:

Bot po infekcji systemu ofiary odnajdywał unikalny ciąg bajtów pozwalający na jednoznaczną identyfikację systemu, np. GUID, przekazując go następnie botmasterowi za pośrednictwem kanału C&C. Botmaster generował unikalną parę kluczy i przypisywał do nich unikalny identyfikator, część publiczną przekazując za pośrednictwem kanału C&C do bota, który używał jej do szyfrowania danych. Po zaszyfrowaniu danych, poinformowaniu ofiary i wpłaceniu okupu, botmaster na podstawie podanego przez ofiarę identyfikatora udostępnił prywatną część klucza, pozwalającą ofierze na odszyfrowanie danych. Autorzy Cry wprowadzili nowy schemat, który umożliwił im zminimalizowanie interakcji z systemem ofiary, jednocześnie nie rezygnując z generowania indywidualnych par kluczy asymetrycznych.



Najpierw bot generuje indywidualną parę kluczy asymetrycznych (na potrzeby tego raportu nazywaną parą operacyjną). Część publiczna jest używana do szyfrowania danych w systemie ofiary, prywatna zaś jest szyfrowana zakodowaną w próbce częścią publiczną drugiej pary kluczy (zwanej parą główną). Zasyfrowany klucz jest dodatkowo przekształcany w ciąg znaków, który stanowi identyfikator ofiary.

By lepiej zrozumieć stosowany schemat wyobraźmy sobie części publiczne kluczy jako kłódki, zaś części prywatne jako klucze do nich. Załóżmy, że pary operacyjne są koloru zielonego, a pary główne – niebieskiego. Posługując się tą reprezentacją, można powiedzieć, że autorzy Cry za pomocą swojego programu dostarczają do systemu ofiary niebieską kłódkę. Po uruchomieniu programu, generuje on indywidualną zieloną kłódkę i zielony klucz. Zielona kłódka jest używana do zamknięcia danych ofiary, a niebieska kłódka jest używana do zamknięcia zielonego klucza.



Zielona kłódka jest inna dla każdej ofiary, tak więc po uzyskaniu swojego zielonego klucza, jedna ofiara nie może przekazać go do wykorzystania innej. Aby uzyskać dostęp do swojego zielonego klucza, ofiara musi wysłać zamkniętą niebieską kłódką zielony klucz do botmastera przy wpłaceniu okupu. Używając swojego niebieskiego klucza botmaster otwiera i odsyła ofierze zielony klucz.

Kanał C&C

Pomimo tego, że wymagania komunikacji pomiędzy botem i botmasterem są ograniczone do minimum, oprogramowanie to posiada jednak dość rozbudowany kanał C&C, składający się z dwóch podkanałów.

Podkanał 1

Opiera się na protokole UDP. W trakcie wykonywania operacji, próbka formułuje raporty, które wysyła za pomocą tego podkanału do ponad 1000 systemów o numerach IP wygenerowanych za pomocą wbudowanego algorytmu będącego odpowiednikiem DGA (Domain Generation Algorithm). Dzięki takiemu rozwiązaniu, zainfekowane systemy ofiar generują dużą ilość ruchu maskującego, który utrudnia zlokalizowanie odbiorcy kanału C&C. Ponieważ protokół UDP jest bezpołączeniowy, można jedynie spekulować, że co najmniej jeden z wygenerowanych adresów IP należy do infrastruktury botmastera. Aby to jednak zweryfikować, należałoby dotrzeć i przeanalizować każdy z nich.

Niewykluczone również, że żaden z wygenerowanych adresów nie należy do botmastera, a generowany ruch ma na celu jedynie zmylenie służb zajmujących się ściganiem tego typu przestępstw.

Podkanał 2

Podkanał 2 to serwis w sieci TOR, w praktyce jedyny niezbędny kanał komunikacyjny. Służy on do przekazania okupu i identyfikatora ofiary botmasterowi i w odpowiedzi do uzyskania od niego klucza deszyfrującego. Dokładny opis, jak korzystać z tego kanału, znajduje się w notatce umieszczonej na pulpicie ofiary.

Można jedynie spekulować na temat przyczyn umieszczenia w próbce aż dwóch podkanałów C&C. Dokładne przeanalizowanie zakończeń podkanału 1 wymagałoby akwizycji dowodów z ponad 1000 systemów, a nawet to nie gwarantuje, że umożliwiłyby one zlokalizowanie infrastruktury botmastera. Prawdopodobną możliwością jest zbieranie danych na temat ofiar, które nie zdecydowały się na kontakt w celu uiszczenia okupu lub złośliwe oprogramowanie odstąpiło od procesu infekcji. Więcej informacji o możliwości odstąpienia od infekcji znajduje się w dalszej części raportu.

Inne uwagi

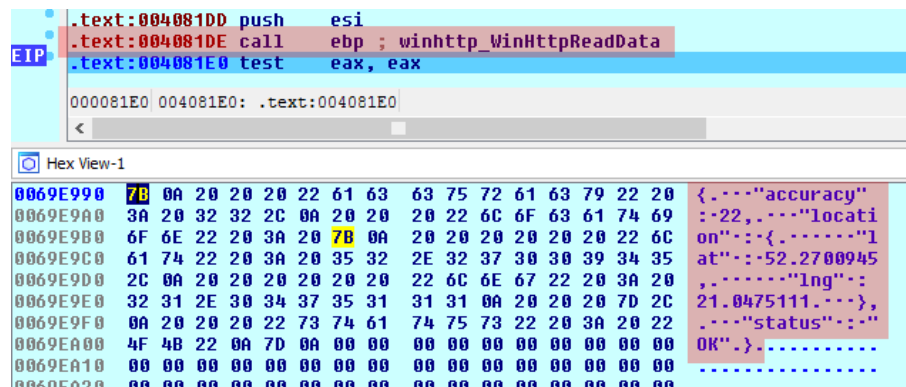
Cry posiada kilka elementów, które odróżniają go od innych programów tej klasy.

Generowanie nazw

Podczas działania Cry tworzy szereg wspierających jego operacje obiektów. Mogą one zostać wykorzystane do wykrycia infekcji (ang. IoC – Indicators of Compromise). Aby tego uniknąć, w próbkę zaimplementowano funkcję generowania pseudolosowych ciągów znaków, wykorzystywanych następnie do tworzenia obiektów. Jako dane wejściowe generatora wprowadzany jest wygenerowany unikalny GUID infekowanego systemu, co zapewnia, że dla każdej infekcji nazwy będą się różniły.

Analiza WLAN

Próbka posiada funkcję badania, czy system ofiary jest podłączony do sieci bezprzewodowych, by pobrać informację o znajdujących się w zasięgu sieciach radiowych, a następnie spróbować połączyć się z API Google Maps w celu geolokalizacji systemu ofiary.



```

.text:004081DD push esi
.text:004081DE call ebp ; winhttp_WinHttpRequestReadData
EIP .text:004081E0 test eax, eax

000081E0 004081E0: .text:004081E0
<

Hex View-1
0069E990 7B 0A 20 20 20 22 61 63 63 75 72 61 63 79 22 20 {..-"accuracy"
0069E9A0 3A 20 32 32 2C 0A 20 20 20 22 6C 6F 63 61 74 69 :-22,.."locati
0069E9B0 6F 6E 22 20 3A 20 7B 0A 20 20 20 20 20 20 22 6C on":-{.."l
0069E9C0 61 74 22 20 3A 20 35 32 2E 32 37 30 30 39 34 35 at":-52.2700945
0069E9D0 2C 0A 20 20 20 20 20 20 22 6C 6E 67 22 20 3A 20 ,.."lng":
0069E9E0 32 31 2E 30 34 37 35 31 31 31 0A 20 20 20 7D 2C 21.0475111.},
0069E9F0 0A 20 20 20 22 73 74 61 74 75 73 22 20 3A 20 22 ,.."status":
0069EA00 4F 4B 22 0A 7D 0A 00 00 00 00 00 00 00 00 00 OK".}.....
0069EA10 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
0069FA00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

```

Po uzyskaniu tej informacji, wysyła ją w formie raportu do podkanału 1 C&C.

Odstąpienie od infekcji

Cry posiada wbudowany mechanizm zabezpieczający, który przewiduje odstąpienie od infekcji ofiary w przypadku spełnienia pewnych warunków. Przed podjęciem decyzji o infekcji pobierane są dane na temat używanych języków klawiatury.


```

.text:00402E90 push    ecx
.text:00402E99 push    0Ah
.text:00402EAB call   ds:user32_GetKeyboardLayoutList
.text:00402EB1 test   eax, eax
.text:00402EB3 jle   short loc_402EF8
.text:00402EB5 xor    edx, edx
.text:00402EB7 test   eax, eax
.text:00402EB9 jle   short loc_402EF8
.text:00402EBB jmp   short loc_402EC0
.text:00402EBD ; -----
.text:00402EBD lea   ecx, [ecx+0]
.text:00402EC0
.text:00402EC0 loc_402EC0:
.text:00402EC0
.text:00402EC0 mov    cx, [esp+edx*4+2Ch+var_28]
.text:00402EC5 mov    esi, 3FFh
.text:00402ECA and   cx, si
.text:00402ECD movzx  ecx, cx
.text:00402ED0 cmp    ecx, 23h
.text:00402ED3 jz    short loc_402EFF
.text:00402ED5 cmp    ecx, 3Fh
.text:00402ED8 jz    short loc_402EFF
.text:00402EDA cmp    ecx, 19h
.text:00402EDD jz    short loc_402EFF
.text:00402EDF cmp    ecx, 22h

```

Jeśli system wykorzystuje język białoruski, kazachski, rosyjski, ukraiński, uzbecki lub estoński, program decyduje się jedynie na wysłanie raportu o podwyższeniu uprawnień w systemie do podkanału 1 C&C, ale odstępuje od szyfrowania danych na dyskach i usuwa sam siebie.

Protektor

Właściwy kod aplikacji Cry jest zabezpieczony przed analizą. Zanim nastąpi jego wykonanie, konieczne jest jego odbezpieczenie. Protektor wykorzystywany w przypadku niniejszej próbki nie odbiega znacząco od innych próbek złośliwego oprogramowania. W pierwszym kroku importowane są wywołania biblioteczne umożliwiające alokację pamięci i rozpakowanie właściwego kodu. Do zaalokowanych fragmentów przenoszony jest, a następnie wykonywany przetwarzany kod.

Jedną z ciekawych konstrukcji jest funkcja modyfikująca własny adres powrotu na podstawie zadanych argumentów.

```

.text:0040C0A5 add    eax, ecx
.text:0040C0A7 ; ===== SUBROUTINE =====
.text:0040C0A7
.text:0040C0A7 sub_40C0A7 proc near
.text:0040C0A7
.text:0040C0A7 arg_0= dword ptr 4
.text:0040C0A7
.text:0040C0A7 push  ecx
.text:0040C0A8 push  edx
.text:0040C0A9 mov    edx, [esp+8+arg_0]
.text:0040C0AB mov    ecx, [esp+8]
.text:0040C0AD add    ecx, 0FFh
.text:0040C0B7 sub    ecx, edx
.text:0040C0B9 inc    ecx
.text:0040C0BB inc    ecx
.text:0040C0BD mov    [esp+8], ecx
.text:0040C0BF pop    edx
.text:0040C0C1 pop    ecx
.text:0040C0C3 retn  4
.text:0040C0C5 sub_40C0A7 endp

```

EAX	00000000
EBX	00000000
ECX	00403F9E
EDX	00008536
ESI	00221EC7
EDI	00000000
EBP	0012FD44
ESP	0012FBE0
EIP	0040C0BB
EFL	00000202

Stack view	
0012FBE0	000030C2
0012FBE4	0000001D
0012FBE8	0040C3D3
0012FBEC	00008536
0012FBF0	002283B8
0012FBF4	50000163
0012FBF8	0000001A
0012FBFC	00228460

Po wywołaniu tej funkcji z określonymi argumentami pobiera ona ze stosu adres powrotu i wykonuje na nim działania arytmetyczne, a wynikiem nadpisuje adres powrotu.

<pre> .text:0040C0A5 add eax, ecx .text:0040C0A7 ; ----- SUBROUT .text:0040C0A7 .text:0040C0A7 .text:0040C0A7 sub_40C0A7 proc near .text:0040C0A7 .text:0040C0A7 .text:0040C0A7 arg_0= dword ptr 4 .text:0040C0A7 .text:0040C0A7 push ecx .text:0040C0A8 push edx .text:0040C0A9 mov edx, [esp+8+arg_0] .text:0040C0AD mov ecx, [esp+8] .text:0040C0B1 add ecx, 0FFh .text:0040C0B7 sub ecx, edx .text:0040C0B9 inc ecx .text:0040C0BA inc ecx .text:0040C0BB mov [esp+8], ecx .text:0040C0BF pop edx .text:0040C0C0 pop ecx .text:0040C0C1 retn 4 .text:0040C0C1 sub_40C0A7 endp </pre>	<pre> EAX 00000000 EBX 00000000 ECX 00403F9E EDX 00000536 ESI 00221EC7 EDI 00000000 EBP 0012FD44 ESP 0012FBE0 EIP 0040C0BF EFL 00000202 </pre>																
	<p>Stack view</p> <table border="1"> <tr><td>0012FBE0</td><td>000030C2</td></tr> <tr><td>0012FBE4</td><td>0000001D</td></tr> <tr><td>0012FBE8</td><td>00403F9E</td></tr> <tr><td>0012FBEC</td><td>00000536</td></tr> <tr><td>0012FBF0</td><td>00228388</td></tr> <tr><td>0012FBF4</td><td>50000163</td></tr> <tr><td>0012FBF8</td><td>0000001A</td></tr> <tr><td>0012FBFC</td><td>00228460</td></tr> </table>	0012FBE0	000030C2	0012FBE4	0000001D	0012FBE8	00403F9E	0012FBEC	00000536	0012FBF0	00228388	0012FBF4	50000163	0012FBF8	0000001A	0012FBFC	00228460
0012FBE0	000030C2																
0012FBE4	0000001D																
0012FBE8	00403F9E																
0012FBEC	00000536																
0012FBF0	00228388																
0012FBF4	50000163																
0012FBF8	0000001A																
0012FBFC	00228460																

W ten sposób funkcja, zamiast autonomicznej jednostki wykonywanych operacji, spełnia rolę skomplikowanej instrukcji skoku do innej części kodu. Takie zastosowanie funkcji może posłużyć do zmylenia narzędzi, które zakładają tradycyjne wykorzystanie instrukcji call.

Odzyskiwanie plików

Ze względu na błędy popełniane w procesie szyfrowania, pliki zaszyfrowane z wykorzystaniem analizowanej próbki są możliwe do odzyskania w drodze analizy z zakresy informatyki śledczej. Użytkownik, który padł ofiarą aplikacji Cry może spróbować odzyskać te dane samodzielnie (na przykład korzystając z darmowego pakietu Sleuthkit przeznaczonego dla systemu Linux) lub skorzystać z usług jednej z kilku firm zajmujących się odzyskiwaniem danych.